# Lab 3 Report

ECE 332 Winter 2020

Jose Manuel Lopez Alcala & Benjamin Carlson

# 1 Introduction

For this week's lab, we built upon the motor speed controller we constructed from lab 2 to create a closed loop speed controller that would self-adjust the speed of the motor as determined by the user. The design takes a desired motor speed from a user as an input, and outputs the corresponding average DC voltage required for the motor to rotate at the desired velocity.

# 2 Methods

## 2.1 Hardware

The hardware was connected almost identically as in the other lab. The main difference here is that we connected the the motor to the controlling board instead of connecting it directly to the power generator. Figure 1 shows the board to which we connected the motor to.

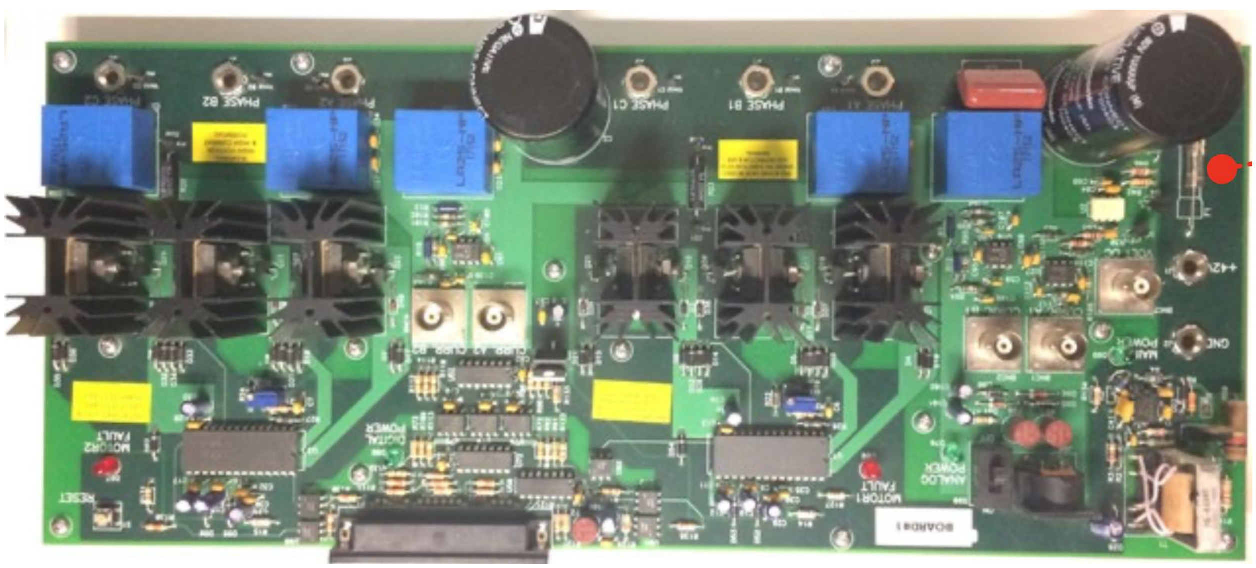


Figure 1. Controlling board for motor

The board was connected to the power supply that supplied the motor power. Additionally, it had another power supply that powered the logic on the board itself.  This power supply is shown in Figure 2.

Figure 2. Power supply for logic on controlling board.

The circuit diagram in Figure 3 describes the logic of the h-bridge circuit. During the A cycle, the A pwm goes high which turns on Q1, while B is low which turns on Q4 and gives the current a path to ground through the motor, as well as a +42V drop causing it to spin in the positive (counter-clockwise) direction. Conversely, during the B cycle, B outputs high and A outputs low, turning on Q3 and Q1, giving an equal and opposite current path and voltage drop, causing rotation in the opposite direction. By varying the cycle time of the pwm output we can alternate the average voltage and control the speed and direction of the motor.

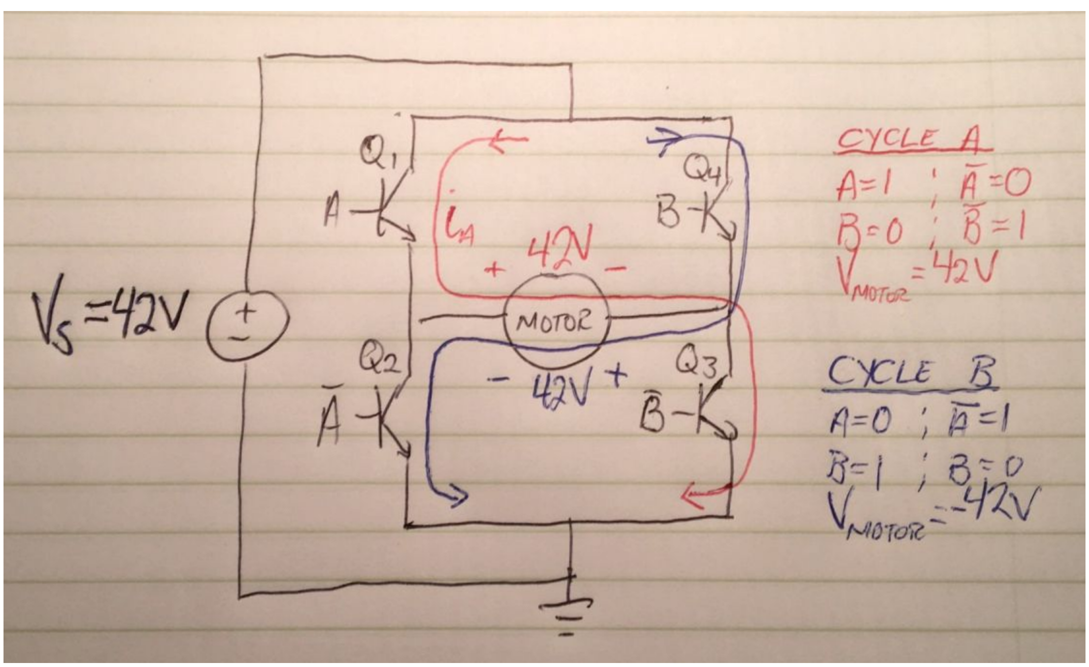


Figure 3. H-bridge circuit.

Building on the original design, we modified the existing schematic to allow for a closed loop system. This was implemented by the addition of an integrated MATLAB function block that took in one input, the desired motor RPM’s and output a voltage to regulate the speed of the motor in a continuous closed loop.

## 2.2 Programming

The change in our design consisted of a moving average that smoothed out the signal and returned an average RPM. The other thing that we added was the Phase controller. This controller block interfaced with dSpace and allowed us to control the speed based on the input of the the CMD block.

With the addition of the MATLAB function program (Figure 6) within the Simulink design, the values of the desired reference velocity (ωref) and the average velocity (ωrpm) were compared. if ωrpm was determined to be greater than ωref (rotating too fast), the voltage was decreased slightly and slowed it down, whereas if rotation was slower than desired, the voltage was increased slightly and consequently sped up. This created a closed loop speed controller that automatically adjusts its speed to the desired user input as illustrated in Figure 4.

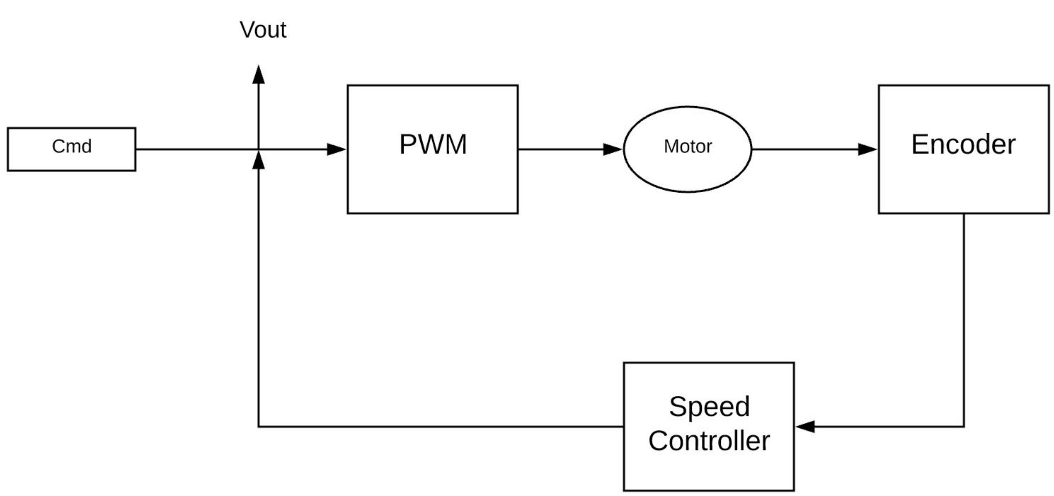


Figure 4. Block diagram of closed loop system. User inputs cmd value between -4000 and +4000 which represents the desired speed of the motor in RPM. The system adjusts PWM voltage through the H-bridge circuit accordingly to maintain the desired speed.

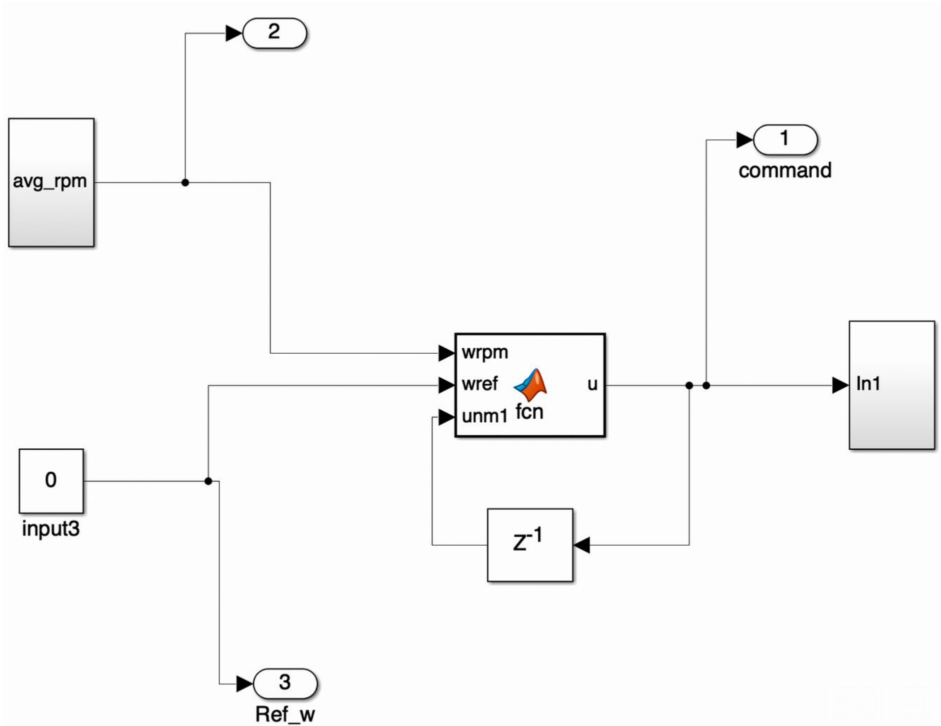

Figure 5. Final closed-loop motor control system design.

We then used the dSpace program for data acquisition and to display the graphs of our outputs as a function of time and thus producing a dynamic model of the system. Furthermore, the dSpace software enabled us to use a slider bar to adjust the desired speed of the system.

```
function u = fcn(wrpm,wref, unm1)

x = wref -wrpm;
if x>0
    u = unm1+0.0001;
elseif x<0
    u = unm1-0.0001;
else
    u = unm1;
end

if u>1
    u=.99;
elseif u<-1
    u=-.99;
end

end
```

Figure 6. MATLAB function code where u = motor speed and unm1 = motor speed during previous clock cycle

# 3 Results

The system operated as expected when implemented, with the ωrpm value basically chasing the set ωref value whenever adjustments to ωref were made. As a result of the 0.1ms timestep used in adjustments to the voltage cycle across the motor, there was a slight oscillation in ωref. This effect was not physically observable, but evident in the graphs produced shown in Figure 7.

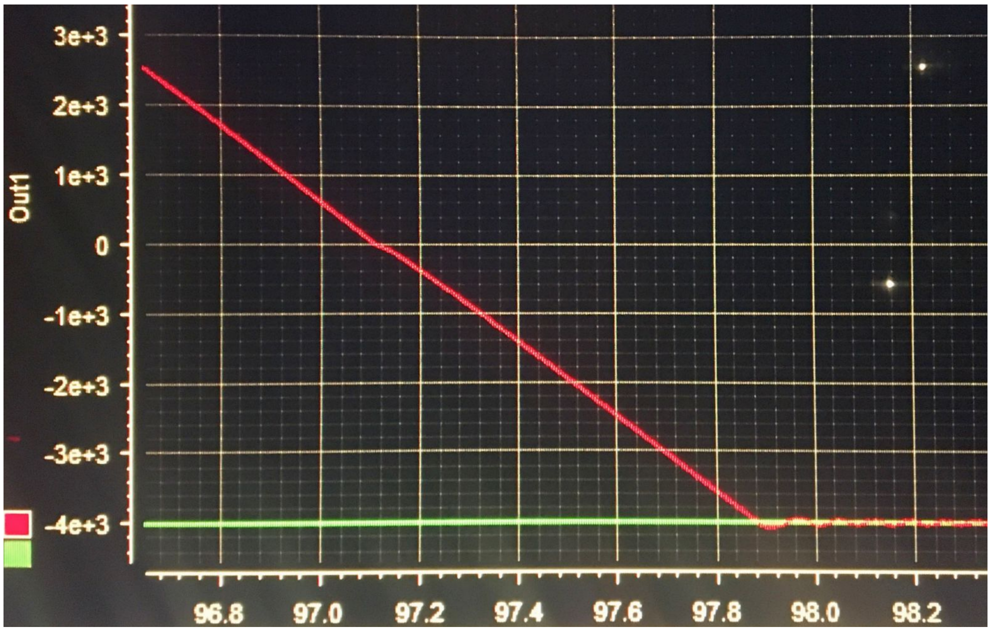


Figure 7. The red line represents ωrpm (actual) and green represents ωref (desired). The snapshot was taken immediately following a rapid decrease in the ωref value. As shown in the graph, ωrpm is steadily decreasing to match ωref. Once it drops below, voltage is increased and ωrpm begins to oscillate slightly, centered around ωref.

# 4 Discussion

While supplying the h-bridge circuit with a constant 42-V DC supply and regulating the cycle duty of the configuration, we were able to control the average dc voltage across the motor, which correlated to being able to control the speed of the motor. Expanding on this system we were able to create a dynamic model a self-adjusting feedback loop using a system identification method within MATLAB to regulate the speed of the motor. Although the system reacted fairly quickly to adjustments made by the user, improvements could be made in decreasing the timesteps between adjustments as well as making finer voltage adjustments when the difference between ωref and ωrpm were very small. Furthermore, by increasing the voltage steps when the difference was very large to reduce lag time. However, for the purpose of this lab, the finished product functioned quite well.